Représentation des entiers signés

Selma - Département Informatique - IUT Fontainebleau/Sénart

Représentation d’un nombre entier en *complément à deux*.

Il s’agit de pouvoir représenter (en binaire) des entiers qui peuvent être positifs ou négatifs.

La longueur du codage est notée k. On dispose donc de 2^k configurations.

Les configurations de 00...0 à 01...1 sont associées respectivement aux entiers de 0 à $2^{k-1}-1$.

Les configurations de 10...0 à 11...1 sont associées respectivement aux entiers -2^{k-1}, $-(2^{k-1}-1)$, …, -1.

$\{0, 1, 2, 3, \ldots, 2^{k-1}-1, 2^{k-1}, 2^{k-1}+1, \ldots, 2^k-4, 2^k-3, 2^k-2, 2^k-1\}$ ← Les nombres modulo 2^k.

Rep(-3)

Rep(-2)

Rep(-1)

Cette manière de représenter les entiers s’appelle la représentation en complément à deux.

Propriétés de la représentation en complément à deux.

→ Une seule représentation pour 0 → 00...0.

→ La représentation d'un nombre positif ou nul commence par 0; celle d'un nombre négatif par 1.

→ Pour tout n non nul dans { $-(2^{k-1}-1)$, …, $2^{k-1}-1$}, Rep(n)+Rep(-n) = 0 mod 2^k où + désigne l'addition binaire.

Conséquence. Si on connaît la représentation Rep(n) de n (positif ou négatif), on obtiendra la représentation Rep(-n) de -n, en inversant tous les bits de Rep(n) et en ajoutant 1 au résultat de cette inversion.

Pour s'en convaincre, on observera que l'addition binaire d'une représentation et de son inverse bit à bit donne le résultat 111..1 (1 partout sur k bits), soit 2^k-1. Pour arriver à 2^k , on ajoute donc 1.

Exemple. Les nombres signés sur k=8 bits, sont :

$-2^7=-128$, $-(2^7-1)=-127$, …-1,0,1, ..., $127=2^7-1$. Ils sont respectivement représentés par :

10000000 , 10000001 , 10000010 , …., 11111111, 00000000 , 00000001 , …, 01111111

Rep(1)+Rep(-1) → 00000001+11111111 = 100000000. Résultat sur 9 bits mais attention, ce n'est pas un débordement. On procède modulo 2^8 !

Exemple. On cherche la représentation de -87 sur 8 bits.

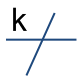
→ On écrit la représentation de 87 sur sur 8 bits : 01010111

→ La représentation de -87 sur sur 8 bits est donc : 10101000 + 1 → 10101001

Comment s’effectue la soustraction sur les entiers par le matériel ?

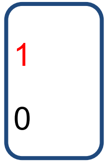
a-b=a+(-b). Ainsi, soustraire un nombre b revient à additionner son opposé.

A l’aide d’une ligne de contrôle spéciale sur 1 bit, appelons-là *inverse,* le matériel se retrouve à faire une soustraction en utilisant le circuit de l’addition.

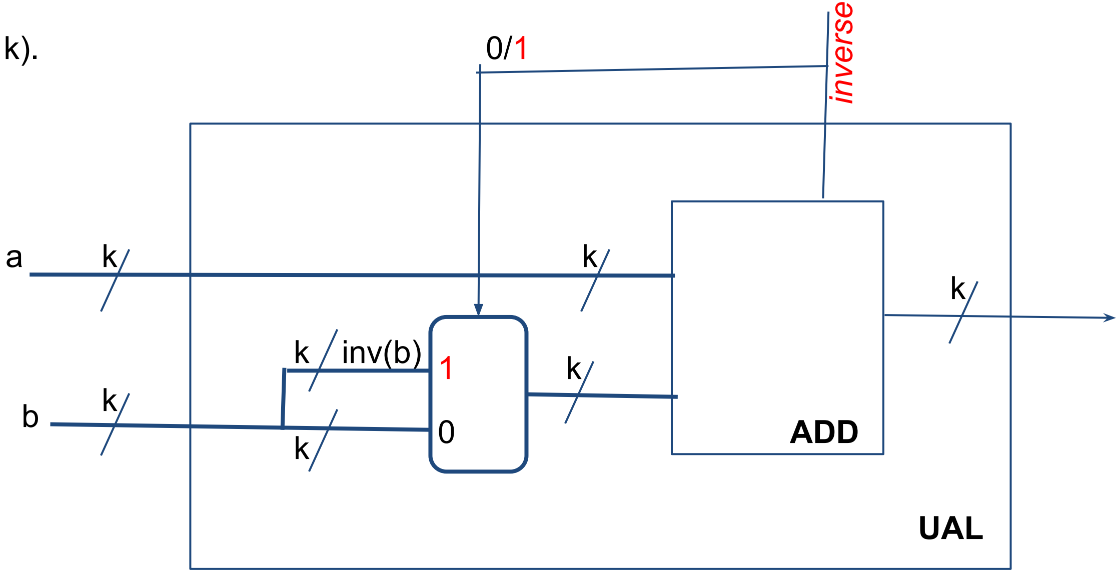


: k fils (la longueur du codage est k).

inv(b) : inverse bit à bit de b.



← Un multiplexeur.
Il recopie sur sa sortie l’entrée dont le numéro est celui de la valeur de la ligne de contrôle.



ADD : Additionneur

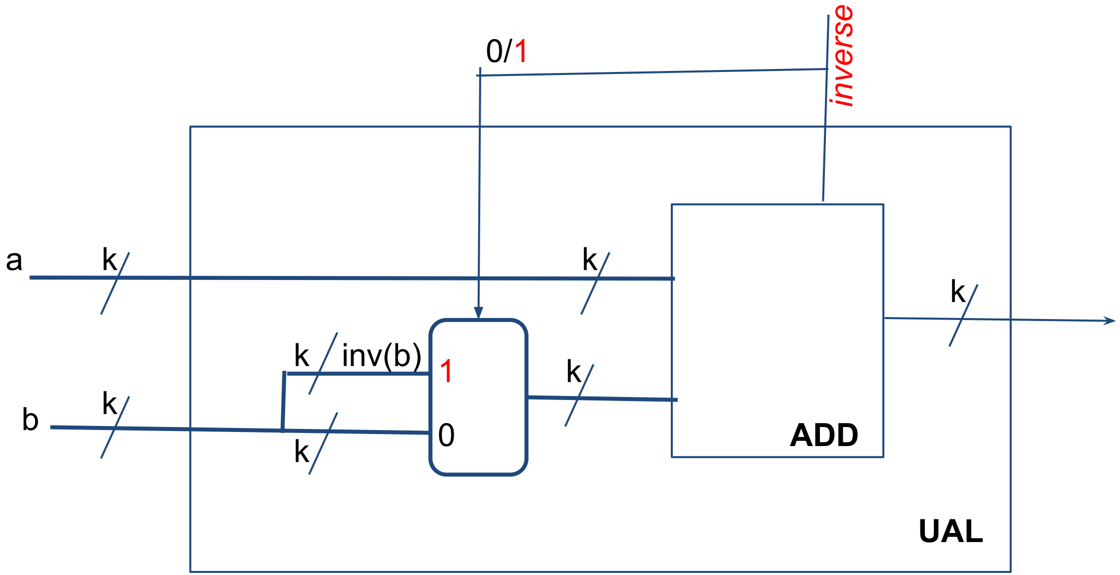
UAL : Unité Arithmétique et Logique (une partie).

La ligne *inverse* est utilisée à la fois comme contrôle du multiplexeur et comme retenue entrante à combiner avec les lignes portant les valeurs des bits de poids faible des opérandes aux entrées de ADD.

Exemple. Sur 8 bits, on donne à faire 12-45.
12: 00001100
45: 00101101
Le matériel fait :
inverse → mis à 1. ADD reçoit alors en entrées a et inv(b) :

inverse : 1

```
    0 0 0 0 1 1 0 0
+   1 1 0 1 0 0 1 0
-------------------
    1 1 0 1 1 1 1 1   ← -33
```



Attention. Une seule addition est faite par le matériel : celle qui ajoute inv(b).

Le débordement (overflow) lors de l'addition des entiers.

On rappelle que sur k bits, les entiers signés représentables sont :

-2^{k-1} , $-(2^{k-1}-1)$, $-(2^{k-1}-2)$, …….. , -1 , 0 , 1 , ………… , $2^{k-1}-1$

10...0 , 10...01 , 10….010 , ……… , 11...1 , 0….0 , 00...01 , …………… , 011..1

Il ne peut pas y avoir de débordement lorsque les opérandes sont de signes opposés : les valeurs absolues sont soustraites l'une de l'autre et pas cumulées.

Le débordement peut survenir lorsque les opérandes sont de même signe car là, les valeurs absolues sont cumulées.

? -2^{k-1} , $-(2^{k-1}-1)$, $-(2^{k-1}-2)$, …….. , -1 , 0 , 1 , ………… , $2^{k-1}-1$

10...0 , 10...01 , 10….010 , ……… , 11...1 , 0….0 , 00...01 , …………… , 011..1

Exemple. Que se passe-t-il lorsque le matériel fait -2^{k-1} +(-3), c’est à dire : 10….0 + 11...101

Signe du débordement : sur les bits de poids fort, les valeurs des retenues en entrée et en sortie sont différentes.

```
      0
      1 0 …..        0 0
+     1 1 1………..1 0 1
______________________
1     0 1 1        1 0 1
```

← Le résultat va être affiché par la machine comme nombre positif alors que les deux opérandes sont négatifs !

-2^{k-1} , $-(2^{k-1}-1)$, $-(2^{k-1}-2)$, …….. , -1 , 0 , 1 , ………… , $2^{k-1}-1$ **?**

10...0 , 10...01 , 10….010 , ……… , 11...1 , 0….0 , 00...01 , …………… , 011..1

De même, lorsqu'un débordement survient lors de l'addition de deux nombres positifs, la machine affiche un résultat négatif.

	1	
	0	……………….
+	0	……………….
0	1	………………..

Signe du débordement : sur les bits de poids fort, les valeurs des retenues en entrée et en sortie sont différentes.

← Le résultat va être affiché par la machine comme nombre négatif alors que les deux opérandes sont positifs !